IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 09/739,618	§	Examiner: Duong, Thomas
Filed: December 18, 2000	§	Group/Art Unit: 2145
Inventor(s):	§	Atty. Dkt. No: 5181-59100
John H. Howard	§	
	§	
	§	
	§	
Title: Object-Based Storage Device with Improved Reliability and Crash Recovery	§	
	§	
	§	
	§	
	§	
	§	

REPLY BRIEF TO EXAMINER'S ANSWER

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Examiner's Answer mailed May 14, 2008 ("Answer" herein), Appellant presents this Reply Brief. Appellant uses the same sections below that were used in the Argument section of the Appeal Brief to respond to section (10), Response to Argument, in the Answer. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

First Ground of Rejection:

Claims 2, 12, 22, and 29:

Appellant respectfully submits that claims 2, 12, 22, and 29 recite combinations of features not taught or suggested in Kozakura. For example, claim 2 recites a combination of features including: "a non-volatile memory storing a first inode locating said first file in said storage and also storing a journal comprising a list of committed inodes; and a block manager … configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory, … record said second inode in said journal".

In the Appeal Brief, Appellant noted that Kozakura's page tables cannot anticipate inodes, contrary to the assertion in the Final Office Action mailed March 21, 2007 ("Office Action"). Page tables locate physical pages stored in the memory system of the computer system, mapping logical pages used by the software to physical pages (see, e.g., Kozakura, col. 1, lines 32-28). The Answer responds (Answer, page 13, first full paragraph), referring to Kozakura's current page table and shadow page table. However, the Answer **fails to address the underlying fact that page tables and inodes are not the same thing**. Accordingly, page tables, no matter how they are constructed or used, cannot anticipate inodes as recited in claim 2.

Appellant also pointed out the current page table data structures cannot be a journal, since they are updated as transactions progress and thus do not comprise a list of committed inodes. The Answer responds to this argument with the assertion that the current page table management table 3 to Applicant's journal (Answer, page 14, lines 2-3). The Answer then cites Kozakura's teachings at col. 4, lines 46-51 and col. 3, lines 34-39 to conclude that the current page table management table 3 anticipates the journal recited in claim 2. See Answer, page 14, first full paragraph. Appellant does not disagree that the current page table management table manages the current and shadow page tables. However, the current and shadow page tables store mappings of virtual page addresses to physical page addresses, and thus are clearly not inodes. Furthermore,

nothing in Kozakura's discussion of the current page table management table 3, the current page tables, and the shadow page tables indicates that the current table management table 3 comprises a list of committed inodes. There are no inodes, and there is no list.

Furthermore, Appellant argued that Kozakura's **checkpoints are not related to a commit command**. **Rather, they are performed cyclically or at a given time** (Kozakura, col. 2, lines 48-49), or **when no transactions are in progress** (Kozakura, col. 2, lines 59-61). **There is no command**. Rather, Kozakura's system creates checkpoints automatically, without any command. The Answer responds with references to the first updating unit and the second updating unit that update the current page table by copying data from the latest page table and updating the copied-to page, and by entering the copied-from page in the shadow page table. (see Answer, page 14, last paragraph extending to page 15). The Answer concludes that the current page table and shadow page table are updated in response to the execution of a transaction. Nevertheless, **execution of a transaction is not a commit command**. Kozakura teaches updating the page tables as a transaction is executed. This has nothing to do with a commit command.

For at least the above stated reasons, Appellant submits that the rejection of claims 2, 12, 22, and 29 is in error and requests reversal of the rejection. The rejection of claims 3-7 (dependent from claim 2), claims 13-20 (dependent from claim 12), claims 23-27 (dependent from claim 22), and claims 30-35 (dependent from claim 29) are similarly in error for at least the above stated reasons, and reversal of the rejection is requested. Each of claims 3-7, 13-20, 23-27, and 30-35 recite additional combinations of features not taught or suggested in the cited art.

Claim 8:

Appellant respectfully submits that claim 8 recites a combination of features not taught or suggested in the cited art. For example, claim 8 recites a combination of features including: "said first inode is stored in an inode file, and wherein said inode file

is identified by a master inode, and wherein said inode file is atomically updated with said second inode by writing said master inode subsequent to said commit command".

As noted in the Appeal Brief, the Office Action alleges that Kozakura's page tables correspond to inodes. However, there is no page table file, identified by a master page table management table, in Kozakura. Therefore, Kozakura cannot anticipate the above highlighted features of claim 8. The Answer asserts that the current page table management table is the master inode (see Answer, page 18, last paragraph extending to page 19). However, the current page table management table is not written anywhere as described in Kozakura, and cannot be a master inode. Furthermore, the current and shadow page tables are not inodes, and the master page table management table is not a master inode nor a journal.

Additionally, Appellant notes that the Answer **inappropriately groups** claim 8 with claims 13 and 30, which are argued separately. Appellant respectfully submits that each argument should be addressed on its own merits.

For at least the above stated reasons, Appellant submits that the rejection of claim 8 is in error and requests reversal of the rejection. The rejection of claims 9-10 (dependent from claim 8) is similarly in error for at least the above stated reasons, and reversal of the rejection is requested. Each of claims 9-10 recite additional combinations of features not taught or suggested in the cited art.

Claims 3 and 23:

Claims 3 and 23 depend from claims 2 and 22, respectively. Accordingly, the rejection of claims 3 and 23 is in error for at least the reasons highlighted above with regard to claims 2 and 22. Additionally, claims 3 and 23 recite combinations of features including: "said commit of said first file comprises a commit command received from an external source which updates said first file."

First, Appellant notes that the Answer **inappropriately groups** claims 3 and 23

with claims 18 and 35, which are argued separately. Appellant respectfully submits that each argument should be addressed on its own merits.

As noted in the Appeal Brief, there is no commit of a file and no commit command in Kozakura. Furthermore, Kozakura does not teach or suggest an external source that updates the page tables, and thus there is no external source as recited in claims 3 and 23. The Answer responds (page 19, last paragraph extending to page 20), once again referring to the first updating unit and the second updating unit making updates to the current and shadow page tables during the execution of a transaction. However, none of these teachings have anything to do with a commit command. Still further, the Answer makes no attempt to identify an external source for a commit command, nor to identify teachings that such an external source updates the first file.

For at least the above stated reasons, Appellant submits that the rejection of claims 3 and 23 is in error and requests reversal of the rejection.

Claims 18 and 35:

Claims 18 and 35 depend from claims 12 and 29, respectively. Accordingly, the rejection of claims 18 and 35 is in error for at least the reasons highlighted above with regard to claims 12 and 29. Additionally, claims 18 and 35 recite combinations of features including: "said establishing said second inode is performed in response to a commit command."

As noted previously, the Answer **inappropriately groups** claims 18 and 35 with claims 3 and 23. Furthermore, the Answer asserts that the first and second updating units modifying the current and shadow pages tables during the execution of a transaction anticipates the above highlighted features. Appellant respectfully submits that nothing in the cited teachings has anything to do with a commit command, and that the pages are updated as the transaction executes, which has nothing to do with committing.

For at least the above stated reasons, Appellant submits that the rejection of

claims 18 and 35 is in error and requests reversal of the rejection.

Claims 13 and 30:

Claims 13 and 30 depend from claims 12 and 29, respectively. Accordingly, the rejection of claims 13 and 30 is in error for at least the reasons highlighted above with regard to claims 12 and 29. Additionally, each of claims 13 and 30 recite a combination of features including: "writing a master inode corresponding to an inode file including said second inode to a checkpoint record in said journal."

As mentioned previously, the Answer **inappropriately groups** claims 13 and 30 with claim 8. The Answer also asserts that the current page table management table corresponds to the master inode (see Answer, page 18, last paragraph extending to page 19). However, the current page table management table is never described as being written anywhere. Certainly, there is no checkpoint record in a journal to which the current page table management table would be written. Accordingly, Kozakura does not teach or suggest the above highlighted features of claims 13 and 30.

For at least the above stated reasons, Appellant submits that the rejection of claims 13 and 30 is in error and requests reversal of the rejection.

Claims 14 and 31:

Claims 14 and 31 depend from claims 13 and 30, respectively. Accordingly, the rejection of claims 14 and 31 is in error for at least the reasons highlighted above with regard to claims 13 and 30. Additionally, each of claims 14 and 31 recite a combination of features including:

scanning said journal to locate a most recent checkpoint record and zero or more inodes subsequent to said most recent checkpoint record within said journal;

copying said master inode from said most recent checkpoint record to a volatile memory; and

updating an inode file corresponding to said master inode with said one or more inodes subsequent to said most recent checkpoint record.

The Office Action cites the same teachings of Kozakura highlighted above with regard to claim 2 to allegedly anticipate the above highlighted features. Appellant respectfully submits that there is no inode file, and no master inode corresponding to that inode file, in Kozakura. Accordingly, Kozakura does not teach or suggest the above highlighted features of claims 14 and 31.

Appellant notes that the Answer does not respond to this argument.

For at least the above stated reasons, Appellant submits that the rejection of claims 14 and 31 is in error and requests reversal of the rejection.

Claims 15 and 32:

Claims 15 and 32 depend from claims 14 and 31, respectively. Accordingly, the rejection of claims 15 and 32 is in error for at least the reasons highlighted above with regard to claims 14 and 30. Additionally, each of claims 15 and 32 recite a combination of features including:

copying one or more blocks of said inode file storing said one or more inodes to a copied one or more blocks; and

updating said master inode in said volatile memory to point to said copied one or more blocks.

The Office Action cites the same teachings of Kozakura highlighted above with regard to claim 2 to allegedly anticipate the above highlighted features. Appellant respectfully submits that there is no inode file, and no master inode corresponding to that inode file, in Kozakura. Accordingly, Kozakura does not teach or suggest the above highlighted features of claims 15 and 32.

Appellant notes that the Answer does not respond to this argument.

For at least the above stated reasons, Appellant submits that the rejection of claims 15 and 32 is in error and requests reversal of the rejection.

Second Ground of Rejection:

Claims 4, 9, 19, and 24:

Claims 4, 9, 19, and 24 depend from claims 3, 8, 18, and 23, respectively. Accordingly, the rejection of claims 4, 9, 19, and 24 is in error for at least the reasons highlighted above with regard to claims 3, 8, 18, and 23. Furthermore, the addition of Fuller to the rejection does not cure the deficiencies in the rejection of claims 3, 8, 18, and 23. Additionally, each of claims 4, 9, 19, and 24 recite a combination of features including: "said commit command comprises a file close command".

As noted in the Appeal Brief, the Office Action cites Fuller to allegedly teach the above features, admitting that Kozakura does not teach the above features. However, Appellant respectfully submits that it would not be obvious to modify Kozakura's page table management system with Fuller's teachings regarding a journaling file system. As highlighted above with regard to claim 2, page tables map logical pages to physical pages in a system memory, and have nothing to do with the file system on a disk. One would simply not be motivated to look to filesystem teachings when considering Kozakura's page table management system for system memory.

As further noted in the Appeal Brief, the Office Action cites the summary of Fuller and Table 1 in Fuller (col. 22, line 35-col. 23, line 23). Appellant respectfully submits that nothing in Fuller's summary teaches or suggests the above highlighted features. Additionally, Fuller describes Table 1 as follows: "Table 1 (hereinafter) illustrates 'commit' and 'NFS commit' assertions for various system calls. Fundamentally using the technique of the present invention, transacted operations will not cause synchronous writes if they do not require a commit and those transacted operations that do require a commit will generate fewer synchronous writes." (Fuller, col. 22, lines 16-21). Table 1 includes a system call that appears to be a file close ("TOP_CLOSE"). However, this call does not cause either type of commit operation to occur. (See the "Commit" and "NFS Commit" columns for TOP_CLOSE in Table 1). Accordingly, even if Kozakura and Fuller were combined as suggested in the Office Action, the combination

would not teach or suggest "said commit command comprises a file close command."

The Answer responds to the above argument (see page 21, first full paragraph), asserting that Fuller teaches transactional commands including 'close', 'fsync', 'read', 'write', 'commit', etc. that can cause the execution of a transaction. Appellant does not disagree that Fuller teaches various commands, but notes that none of these teach a commit command that is a file close command. Indeed, the above list includes separate commit and close commands.

The Answer also cites Fuller, col. 1, lines 51-55. Appellant respectfully submits that Fuller teaches: "The single transaction technique for journaling file systems disclosed herein is of especial utility in overcoming the performance degradation which may be experienced in conventional journaling file systems by entering each file system operation into the current active transaction. Consequently, each transaction is composed of a plurality of file system operations which are then simultaneously committed with a single computer mass storage device disk drive 'write'." (Fuller, col. 1, lines 51-57). Nothing in this section teaches that the commit command is a file close command. Instead, the second teaches that a plurality of file system operations are accumulated and performed as a single write operation.

For at least the above stated reasons, Appellant submits that the rejection of claims 4, 9, 19, and 24 is in error and requests reversal of the rejection.

Claims 5, 10, 20, and 25:

Claims 5, 10, 20, and 25 depend from claims 3, 8, 18, and 23, respectively. Accordingly, the rejection of claims 5, 10, 20, and 25 is in error for at least the reasons highlighted above with regard to claims 3, 8, 18, and 23. Furthermore, the addition of Fuller to the rejection does not cure the deficiencies in the rejection of claims 3, 8, 18, and 23. Additionally, each of claims 5, 10, 20, and 25 recite a combination of features including: "said commit command comprises an fsync command".

The Office Action cites Fuller to allegedly teach the above features, admitting that Kozakura does not teach the above features. The Office Action cites the summary of Fuller and Table 1 in Fuller (col. 22, line 35-col. 23, line 23). However, Appellant respectfully submits that it would not be obvious to modify Kozakura's page table management system with Fuller's teachings regarding a journaling file system. As highlighted above with regard to claim 2, page tables map logical pages to physical pages in a system memory, and have nothing to do with the file system on a disk. One would simply not be motivated to look to filesystem teachings when considering Kozakura's page table management system for system memory.

The Answer responds to the above argument (see page 22, first full paragraph), asserting that Fuller teaches transactional commands including 'close', 'fsync', 'read', 'write', 'commit', etc. that can cause the execution of a transaction. Appellant does not disagree that Fuller teaches various commands, but notes that none of these teach a commit command that is a file close command. Indeed, the above list includes separate commit and fsync commands.

The Answer also cites Fuller, col. 1, lines 51-55. Appellant respectfully submits that Fuller teaches: "The single transaction technique for journaling file systems disclosed herein is of especial utility in overcoming the performance degradation which may be experienced in conventional journaling file systems by entering each file system operation into the current active transaction. Consequently, each transaction is composed of a plurality of file system operations which are then simultaneously committed with a single computer mass storage device disk drive 'write'." (Fuller, col. 1, lines 51-57). Nothing in this section teaches that the commit command is an fsync command. Instead, the second teaches that a plurality of file system operations are accumulated and performed as a single write operation.

For at least the above stated reasons, Appellant submits that the rejection of claims 5, 10, 20, and 25 is in error and requests reversal of the rejection.

Third Ground of Rejection:

Appellant notes that the Answer **inappropriately groups** all of the claims in the third ground together. Appellant respectfully submits that each argument should be decided on its own merits.

Claims 6 and 26:

Claims 6 and 26 depend from claims 2 and 22, respectively. Accordingly, the rejection of claims 6 and 26 is in error for at least the reasons highlighted above with regard to claims 2 and 22. Furthermore, the addition of Zheng to the rejection does not cure the deficiencies in the rejection of claims 2 and 22. Additionally, each of claims 6 and 26 recite a combination of features including: "said journal further includes a checkpoint record including a description of an inode file, a block allocation bitmap, and an inode allocation bitmap."

The Office Action alleges that the above highlighted features are taught in Zheng, specifically citing col. 3, line 3-col. 4, line 14 and col. 14, line 46-col. 15, line 14 of Zheng. Appellant respectfully submits that nothing in these sections teaches or suggests the above highlighted features. While the cited portions of Zheng do generally discuss block allocation and inode allocation, no mention of a checkpoint record, a block allocation bitmap, and an inode allocation bitmap is made.

The Answer responds to this argument by quoting col. 14, lines 46-col. 15, line 14 of Zheng, which appears to be a snippet of code (see Answer, page 22, last paragraph extending to page 23). While it is unclear what the Answer is referring to in this section, Appellant notes that there is only one bitmap referred to in the cited section (the active block bitmap). At the very least, the active block bitmap does not teach two bitmaps. Appellant respectfully submits that at least the inode allocation bitmap is not taught.

For at least the above stated reasons, Appellant submits that the rejection of claims 6 and 26 is in error and requests reversal of the rejection.

Claims 7 and 27:

Claims 7 and 27 depend from claims 6 and 26, respectively. Accordingly, the rejection of claims 7 and 27 is in error for at least the reasons highlighted above with regard to claims 6 and 26. Additionally, each of claims 7 and 27 recite a combination of features including: "the description comprises inodes for each of said inode file, said block allocation bitmap, and said inode allocation bitmap."

The same sections of Zheng highlighted above with regard to claims 6 and 26 are cited to allegedly teach the features of claims 7 and 27. However, these sections do not teach or suggest the block allocation bitmap and the inode allocation bitmap, as highlighted above. Furthermore, Zheng does not teach or suggest inodes for the block allocation bitmap and the inode allocation bitmap, either.

Appellant notes that the Answer does not respond to this argument.

For at least the above stated reasons, Appellant submits that the rejection of claims 7 and 27 is in error and requests reversal of the rejection.

Claims 16 and 33:

Claims 16 and 33 depend from claims 12 and 29, respectively. Accordingly, the rejection of claims 16 and 33 is in error for at least the reasons highlighted above with regard to claims 12 and 29. Furthermore, the addition of Zheng to the rejection does not cure the deficiencies in the rejection of claims 12 and 29. Additionally, each of claims 16 and 33 recite a combination of features including:

> said block map further comprises a first inode allocation bitmap indicating which inodes within said first inode file are allocated to files, the method further comprising:
>
> > copying said first inode allocation bitmap to a second inode allocation bitmap;
> >
> > modifying said second inode allocation bitmap to reflect one or more inodes allocated to new files; and

establishing a third inode within said block map to said second inode allocation bitmap subsequent to said modifying said second inode bitmap.

The Office Action alleges that the above highlighted features are taught in Kozakura and Zheng, citing the same teachings of Kozakura and Zheng used in the rejections of the other claims. Appellant respectfully submits that nothing in these sections teaches or suggests the above highlighted features. While the cited portions of Zheng do generally discuss block allocation and inode allocation, no mention of a an inode allocation bitmap is made, nor any copying of inode allocation bit maps and modifying of the copied bitmaps.

Appellant notes that the Answer does not respond to this argument.

For at least the above stated reasons, Appellant submits that the rejection of claims 16 and 33 is in error and requests reversal of the rejection.

Claims 17 and 34:

Claims 17 and 34 depend from claims 16 and 33, respectively. Accordingly, the rejection of claims 17 and 34 is in error for at least the reasons highlighted above with regard to claims 16 and 33. Additionally, each of claims 17 and 34 recite a combination of features including:

a first block allocation bitmap indicating which blocks within a storage including said block map are allocated to files, the method further comprising:

copying said first block allocation bitmap to a second block allocation bitmap;

modifying said second block allocation bitmap to reflect one or more blocks allocated to files; and

establishing a fourth inode within said block map to said second block allocation bitmap subsequent to said modifying said

second block allocation bitmap.

The Office Action alleges that the above highlighted features are taught in Kozakura and Zheng, citing the same teachings of Kozakura and Zheng used in the rejections of the other claims. Appellant respectfully submits that nothing in these sections teaches or suggests the above highlighted features. While the cited portions of Zheng do generally discuss block allocation and inode allocation, no mention of a an block allocation bitmap is made, nor any copying of block allocation bit maps and modifying of the copied bitmaps.

Appellant notes that the Answer does not respond to this argument.

For at least the above stated reasons, Appellant submits that the rejection of claims 17 and 34 is in error and requests reversal of the rejection.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejections of claims 2-10, 12-20, 22-27, and 29-35 are erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge any other fees that may be due, or credit any overpayment, to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-59100/LJM.

Respectfully submitted,

/Lawrence J. Merkel/
Lawrence J. Merkel, Reg. No. 41,191
Agent for Appellant

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: July 14, 2008